Dex1B: Learning with 1B Demonstrations for Dexterous Manipulation

Author Names Omitted for Anonymous Review. Paper-ID [77]

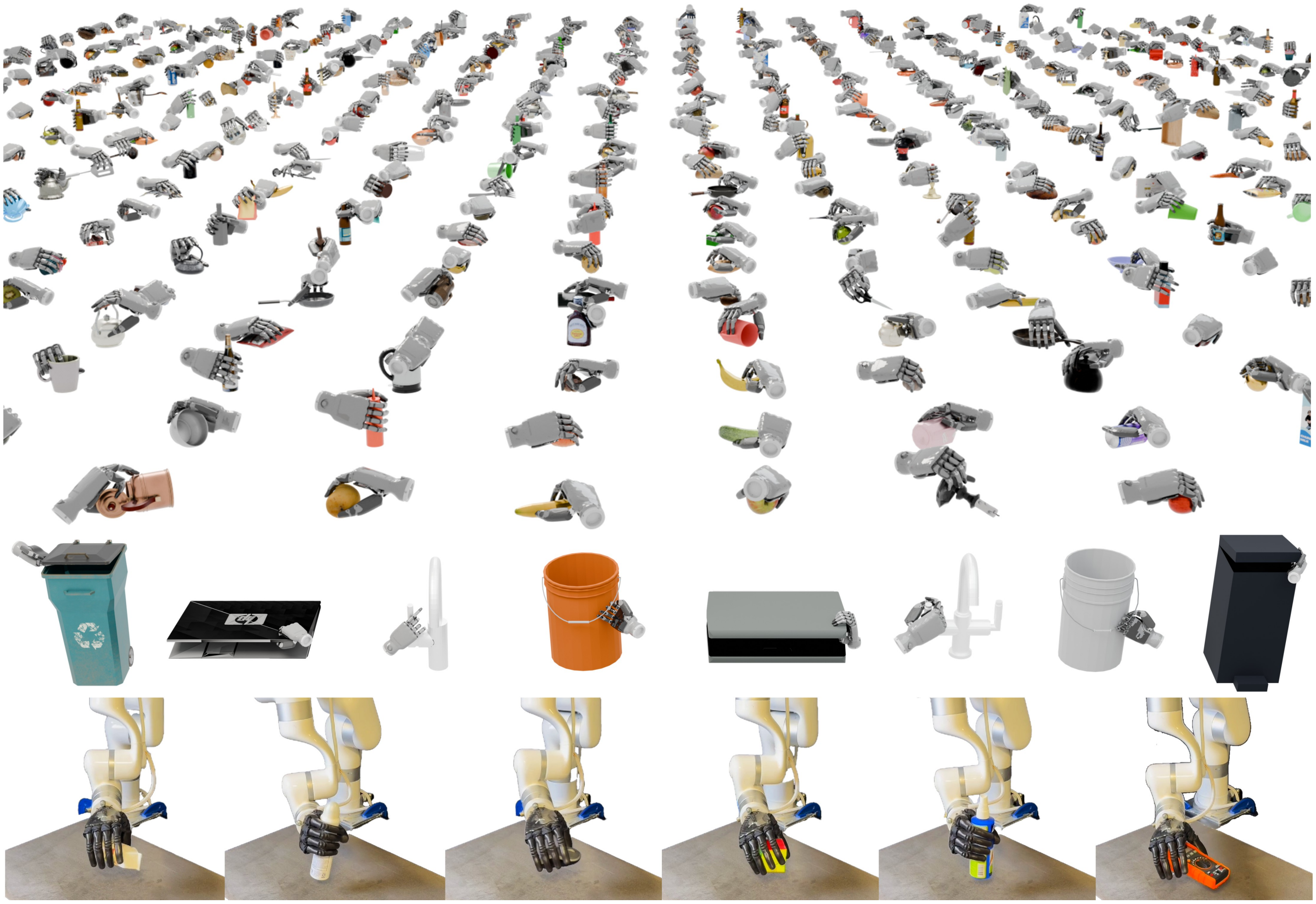

Fig. 1: The **Dex1B** benchmark consists of 1B generated high-quality demonstrations for grasping (top) and articulation (middle) tasks. At the bottom, we show the **direct sim-to-real** transfer results of our method **DexSimple** trained with Dex1B. This demonstrates that Dex1Bis both scalable and generalizable to real environments.

***Abstract*—Generating large-scale demonstrations for dexterous hand manipulation remains challenging, and several approaches have been proposed in recent years to address this. Among them, generative models have emerged as a promising paradigm, enabling the efficient creation of diverse and physically plausible demonstrations. In this paper, we introduce Dex1B, a large-scale, diverse, and high-quality demonstration dataset produced with generative models. The dataset contains one billion demonstrations for two fundamental tasks: grasping and articulation. To construct it, we propose a generative model that integrates geometric constraints to improve feasibility and applies additional conditions to enhance diversity. We validate the model on both established and newly introduced simulation benchmarks, where it significantly outperforms prior state-of-the-art methods. Furthermore, we demonstrate its effectiveness and robustness through real-world robot experiments.**

I. Introduction

Dexterous manipulation with hand has been a long-standing topic in robotics. While its highly flexible and dynamic nature allows for more complex and robust manipulation skills, the high degrees of freedom (DoF) of a hand makes it very challenging to achieve its ideal function. In fact, with recent advancements in applications using parallel-jaw grippers [24, 9, 6, 25, 1], researchers in the community have started questioning the necessity of dexterous hands and having doubts about whether hands are only making problems harder.

We argue that dexterous hand is indeed valuable, but we just did not have enough data to capture the diverse

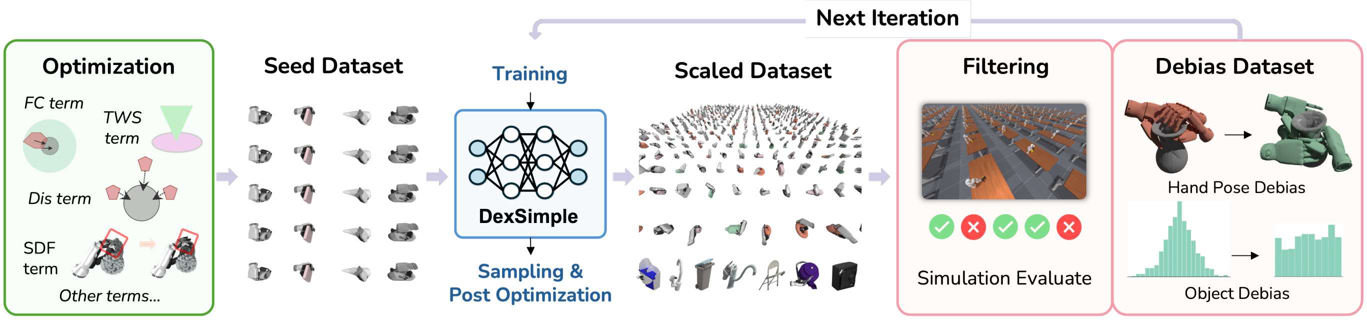

Fig. 2: ***Dex1B*** demonstration collection. The engine takes object assets and hand pose initialization as input, using a control-based optimization algorithm to generate the Seed dataset. Then the Seed dataset is used as the training data for DexSimple, else for Dex1Bfor the last iteration. Then DexSimple will generate a scaled proposal dataset with π as the scaling ratio. For the proposal dataset, we then use the simulation critic and debiased algorithm to create the debiased dataset for optimization refinement.

and complex distributions required for effective dexterous manipulation. To address this data scarcity, previous works have explored various approaches, including human demonstrations [13, 16, 5, 4], optimization-based methods [19, 3, 18], reinforcement learning (RL)-based techniques [7, 26]. While these methods help generate demonstrations at a certain scale, they each have limitations: human annotation is costly and imprecise, optimization-based methods are slow and sensitive to initialization, and RL-based techniques lack data diversity.

Meanwhile, a large body of generative models [10, 14, 23, 12, 20] has been proposed in recent years to model the distribution of demonstration datasets, motivating us to explore how generative models can be leveraged for data generation. And we identify two key issues when applying generative models on data generation: i). **Feasibility:** The success rate of generative models is often lower than that of discriminative models. ii). **Diversity:** While generative models can produce more diverse actions than discriminative models, they still tend to interpolate between the seen demonstrations, which may maintain or even reduce the original level of diversity of whole dataset rather than expanding it.

To address the feasibility issue, we propose incorporating geometric constraints into the generative model, which significantly improves its performance. We also integrate optimization techniques with generative models, leveraging the strengths of both approaches: optimization ensures physically plausible results, while generative models enable efficient large-scale generation. To improve diversity, we introduce additional conditions to the generative model and prioritize sampling actions for less frequent condition values, encouraging the model to generate actions that differ more from existing ones in the dataset.

Our approach begins with an optimization-based method to construct a small yet high-quality seed dataset of dexterous manipulation demonstrations. We then train a generative model on this dataset and use it to scale up data generation efficiently. To mitigate biases introduced by optimization, we propose an debias mechanism, which systematically improves the diversity of generated data. This framework results in Dex1B, a dataset comprising one billion dexterous hand demonstrations, representing a substantial advancement in scale, diversity and quality over existing datasets. Compared to DexGraspNet [19], which operates on the object set of similar scale, our dataset offers 700× more demonstrations, significantly enriching the available training data for learning-based models. Unlike previous approaches that rely solely on human annotation or optimization, our method combines optimization and neural networks, achieving a superior balance between cost, efficiency, and data quality.

To effectively leverage the scale and diversity of Dex1B, we introduce DexSimple, a new baseline that extends prior work [10] by incorporating conditional generation and enhanced loss functions. Despite its simplicity, DexSimple benefits from the scale and diversity of Dex1B, achieving state-of-the-art (SoTA) performance across dexterous manipulation tasks.

II. Dex1B Benchmark

We introduce a comprehensive benchmark for two fundamental dexterous manipulation tasks: **grasping** and **articulation**. In the grasping task, the robot hand must reach for and lift an object, whereas the articulation task requires the hand to manipulate an articulated object to achieve a specific degree of opening. Our benchmark consists of over 6,000 diverse objects and provides one billion demonstrations across three dexterous hands: the Shadow Hand, the Inspire Hand, and the Ability Hand. Each demonstration consists of a complete action sequence, from initial reaching to object manipulation.

To generate these demonstrations, we synthesize key hand poses at critical interaction points with the object, while the remaining action sequences—such as reaching, lifting, and opening—are generated using motion planning. The evaluation of our benchmark is conducted with ManiSkill [17, 21].

Overview of Data Generation. Broadly, hand pose generation for dexterous manipulation can be approached through optimization-based methods or generative models. While optimization methods can be effective, they are often computationally expensive, especially for large-scale generation, and tend to bias the dataset toward simpler cases. On the other hand, generative models rely on an initial dataset to learn meaningful data distributions. In this work, we integrate both approaches to leverage their strengths.

As illustrated in Figure 2, we begin by constructing a small-scale seed dataset using optimization. This seed dataset serves as the foundation for training a generative model to learn its underlying data distribution. The trained generative model is then used to produce additional demonstrations. However, since the generative model inherently inherits the biases of the seed dataset, we introduce a debiasing strategy to enhance diversity. Specifically, we condition the generative model on targeted factors to generate hand poses under less frequently observed conditions, thereby expanding the dataset beyond the initial distribution. By iteratively refining the generative model through repeated training and debiasing operations, we construct our final dataset, Dex1B, which achieves both diversity and robustness in dexterous manipulation demonstrations.

Optimization for Seed Dataset. To generate the seed dataset, we implement an efficient optimization method for hand pose synthesis based on previous work [19, 3], while including new features like scene-level collision avoidance and support for various hands. Although the optimization process is well-engineered (1,000 grasps per minutes on a single GPU), generating one billion demonstrations remains computationally expensive. Therefore, we only use optimization to create a small-scale seed dataset (around 5 million poses).

Generative Models for Scaling-up Demonstrations. Generative models are widely adopted for capturing the distribution of action demonstrations. However, applying these models for data generation still presents several challenges: i). **Feasibility:** The success rate of generative models is often lower than that of discriminative models, leading to a higher proportion of infeasible samples. ii). **Limited Diversity:** While generative models can produce more diverse actions than discriminative models, they still tend to interpolate between the demonstrations, which may maintain or even reduce the original level of diversity of whole dataset rather than expanding it.

To address the feasibility issue, we first incorporate geometric constraints during the generation process, enabling our model to outperform state-of-the-art generative models. In addition, we apply a post-optimization step to the sampled hand poses to prevent penetration and ensure that the fingers closely cover the object.

To improve diversity, we encourage the generative model to sample actions that differ more from existing actions in the dataset while maintaining success rate. To achieve this, we introduce an additional condition to the generative model and prioritize sampling actions for less frequent conditions. Specifically, we associate each hand pose with a single 3D point on the object. We first define the heading direction $v \in \mathbb{R}^3$ of a hand pose as the vector from the palm center to the midpoint between the thumb tip and the middle finger tip. The closest point along this direction is then assigned as the associated point of the hand pose. We adapt our generative model to take the feature vector of a 3D point as a condition for generating corresponding actions. During data generation, we first statistically compute the probability of each point associated with existing actions on the object and then sample new actions inversely proportional to this probability. Additionally, we statistically count the number of existing actions for each object and sample more actions for the more challenging ones.

After increasing the dataset size and diversity, retraining the model on the expanded dataset can further improve its performance. This *iterative data generation* process can be repeated multiple times to progressively refine both the model and the dataset.

III. DexSimple Model

While a large body of generative models [10, 14, 23, 12, 20] have been proposed for dexterous hand manipulation in recent years, their use for data generation or policy deployment remains limited. In this work, we revisit the simple CVAE model and demonstrate that incorporating an SDF-based geometric constraint during training enables it to outperform state-of-the-art methods by a large margin. Furthermore, we integrate additional condition over the base model to support diverse data generation.

Vision Encoder and CVAE. We employ a point cloud $P \in \mathbb{R}^{N\times 3}$ as the visual input, using a full point cloud sampled from the object mesh for data generation and a single-view depth map for policy deployment. We utilize PointNet [15] to encode the point cloud into a global feature vector $f_{\texttt{obj}} \in \mathbb{R}^d$ and local feature vectors $f_p \in \mathbb{R}^d$ for each point $p \in \mathbb{R}^3$:

$$f_{\texttt{obj}}, \{f_p\}_{p\in P} = \textbf{PointNet}(P).$$

The VAE model uses a multi-layer perceptron (MLP) to encode the hand pose g into the mean and standard deviation vectors of a latent distribution. A sample is drawn from this distribution and passed to the MLP decoder to reconstruct the original hand pose. After concatenating conditional vectors (e.g., the global point cloud feature vector $f_{\texttt{obj}}$) to both the inputs of the VAE encoder and decoder, the CVAE model can generate samples under a given condition:

$$\begin{aligned} \mu, \sigma &= \textbf{Enc}(g, f_{\texttt{obj}}), \\ z &= \mu + \sigma \odot \epsilon, \quad \epsilon \sim \mathcal{N}(0, I), \\ \hat{g} &= \textbf{Dec}(z, f_{\texttt{obj}}). \end{aligned} \tag{1}$$

In our work, we simply concatenate additional vectors to incorporate more conditions. Each hand pose is associated with a single object 3D point p by finding the closest point along its heading direction v. To achieve this, we concatenate the corresponding local object feature vector f_p with the CVAE conditional vector.

IV. Experiments

A. Grasping Synthesis Evaluation

Grasping is essential in most manipulation tasks, we firstly evalute the proposed method's effectiveness in grasp synthesis using the DexGraspNet [19] benchmark. We train DexSimple solely with the benchmark's provided training data, reducing

	Setting		Quality			Diversity	
Method	Opt	Filter	SR ↑	Q_1 ↑	Pen ↓	H mean ↑	H std ↓
DDG [11]			67.5	0.058	0.17	5.68	1.99
UGG [14]			43.6	0.026	0.43	8.33	0.30
DexSimple			63.7	0.075	0.29	8.53	0.25
UDG [22]	✓		23.3	0.056	0.15	6.89	0.08
GraspTTA [10]	✓		24.5	0.027	0.68	6.11	0.56
UGG [14]	✓		64.1	0.036	0.17	8.31	0.28
DexSimple	✓		**86.0**	**0.125**	**0.13**	**8.56**	0.15
UGG [14]	✓	✓	72.7	0.063	0.14	7.17	0.07
DexSimple	✓	✓	92.6	0.132	0.12	8.56	0.16

TABLE I: **Grasping synthesis** results on the DexGraspNet [19] benchmark. The proposed generative model, DexSimple, significantly outperforms all baseline methods. Some evaluation results are taken from UGG [14].

		Eval on DexYCB		Eval on Dex1B	
Method	Training Data	Train set	Test set	Train set	Test set
BC w. PointNet	DexYCB [2]	34.72	3.03	1.02	2.56
DexSimple	DexYCB [2]	43.49	21.21	23.68	22.80
BC w. PointNet	Dex1B (ours)	33.02	31.82	31.40	28.54
DexSimple	Dex1B (ours)	**47.17**	**53.02**	**49.58**	**45.40**

(a) Lifting task comparison on DexYCB [2] and Dex1B.

		Eval on ARCTIC		Eval on Dex1B	
Method	Training Data	Train set	Test set	Train set	Test set
BC w. PointNet	ARCTIC [8]	41.03	25.62	37.65	30.16
DexSimple	ARCTIC [8]	48.75	23.08	49.16	51.57
BC w. PointNet	Dex1B (ours)	57.50	63.67	64.74	56.88
DexSimple	Dex1B (ours)	**72.00**	**73.49**	**77.05**	**64.79**

(b) Articulation task comparison on ARCTIC [8] and Dex1B.

TABLE II: Benchmarks on **(a) lifting tasks** with DexYCB [2] and our datasets, and **(b) articulation tasks** with ARCTIC [8] and our datasets. Models trained on Dex1B consistently outperform those trained on DexYCB/ARCTIC across various tasks, baselines, and splits.

the output to a single frame and omitting conditioning during training.

We present quantitative results in Table I. Many grasp generation methods, such as UGG [14], commonly employ post-optimization to enhance performance. To ensure a fair comparison, we indicate the use of post-optimization (abbreviated as "Opt") in the table. The results show that the proposed generative model, DexSimple, outperforms all baseline methods by a large margin. In terms of quality, DexSimple (with post-optimization) achieves the highest success rate (86.0%), the highest Q_1 score (0.125), and the lowest penetration (0.13). For diversity, DexSimple outperforms baseline with a higher entropy mean of 8.56.

UGG [14] proposes a learning-based discriminator to filter grasping, which can be applied to our method. With this filtering, the success rate increases to 92.6%.

B. Benchmarks

We benchmark two methods for grasping and articuation tasks on our datasets, and compare them with the same methods trained on DexYCB [2] and ARCTIC [8]. In addition to the proposed DexSimple, we implement a vanilla behavioral cloning with PointNet [15] (referred to as BC w. PointNet). This model takes the object point cloud, current hand joint values, and poses as input to predict chunked actions for the next $n = 50$ frames. The predicted actions are then merged using a temporal weighting technique form ACT [24].

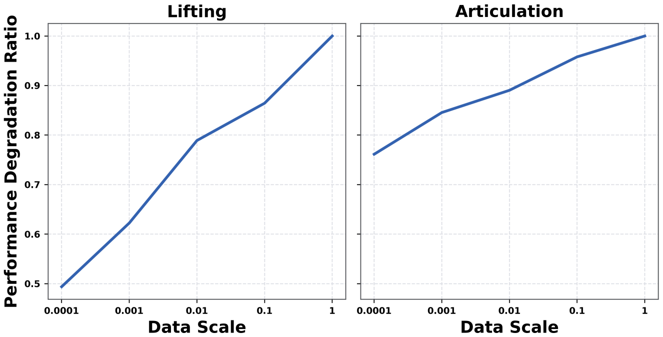

Fig. 3: **Scaling the number** of demonstrations used for training. For both tasks, our model consistently improves with more training data.

The results are reported in Tab. II. When comparing models trained on Dex1B to those trained on DexYCB/ARCTIC, we consistently find that the former outperforms the latter across tasks, baselines and splits. This suggests that supervised learning methods perform better when trained on our larger and more diverse Dex1B dataset. Tab. II also demonstrates that the proposed generative method, DexSimple, achieves better performance than the regression-based BC baselines on both the relatively small DexYCB/ARCTIC dataset and the larger-scale Dex1B. For lifting task, it also can be clearly observed that models trained on DexYCB struggle to generalize to unseen objects.

C. Scaling the Dataset

To investigate the effect of training data size on performance, we reduce the amount of training data and analyze its impact on the success rates of both the lifting and articulation tasks. As shown in Fig. 3, the performance degradation ratio increases as data is reduced, illustrating that the success rates of the proposed DexSimple consistently improve with more training data. Notably, we observe that performance degradation is more pronounced for the lifting task than for the articulation task as training data decreases. We hypothesize this is because lifting relies heavily on stable object grasping, requiring a precise geometric understanding of individual objects, which becomes more challenging with reduced data. In contrast, the articulation task, which emphasizes trajectory execution, shows greater resilience to data reduction as it can adapt to unseen objects through a more generalized approach to motion. This suggests that while both tasks benefit from larger datasets, lifting requires a more extensive dataset to achieve stable performance, whereas articulation maintains reasonable performance even with less data.

References

[1] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. $\pi 0$: A vision-language-action flow model for general robot control. *arXiv*, 2024.

[2] Yu-Wei Chao, Wei Yang, Yu Xiang, Pavlo Molchanov, Ankur Handa, Jonathan Tremblay, Yashraj S Narang, Karl Van Wyk, Umar Iqbal, Stan Birchfield, et al. Dexycb: A benchmark for capturing hand grasping of objects. In *CVPR*, 2021.

[3] Jiayi Chen, Yuxing Chen, Jialiang Zhang, and He Wang. Task-oriented dexterous grasp synthesis via differentiable grasp wrench boundary estimator. *IROS*, 2024.

[4] Yuanpei Chen, Chen Wang, Yaodong Yang, and Karen Liu. Object-centric dexterous manipulation from human motion data. In *CoRL*, 2024.

[5] Zoey Qiuyu Chen, Karl Van Wyk, Yu-Wei Chao, Wei Yang, Arsalan Mousavian, Abhishek Gupta, and Dieter Fox. Dextransfer: Real world multi-fingered dexterous grasping with minimal human demonstrations. *RSS IL workshop*, 2022.

[6] Cheng Chi, Zhenjia Xu, Chuer Pan, Eric Cousineau, Benjamin Burchfiel, Siyuan Feng, Russ Tedrake, and Shuran Song. Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots. In *RSS*, 2024.

[7] Sammy Christen, Lan Feng, Wei Yang, Yu-Wei Chao, Otmar Hilliges, and Jie Song. Synh2r: Synthesizing hand-object motions for learning human-to-robot handovers. In *ICRA*, 2024.

[8] Zicong Fan, Omid Taheri, Dimitrios Tzionas, Muhammed Kocabas, Manuel Kaufmann, Michael J. Black, and Otmar Hilliges. ARCTIC: A dataset for dexterous bimanual hand-object manipulation. In *CVPR*, 2023.

[9] Zipeng Fu, Tony Z. Zhao, and Chelsea Finn. Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation. In *CoRL*, 2024.

[10] Hanwen Jiang, Shaowei Liu, Jiashun Wang, and Xiaolong Wang. Hand-object contact consistency reasoning for human grasps generation. In *ICCV*, 2021.

[11] Min Liu, Zherong Pan, Kai Xu, Kanishka Ganguly, and Dinesh Manocha. Deep differentiable grasp planner for high-dof grippers. In *RSS*, 2020.

[12] Shaowei Liu, Yang Zhou, Jimei Yang, Saurabh Gupta, and Shenlong Wang. Contactgen: Generative contact modeling for grasp generation. In *ICCV*, 2023.

[13] Yumeng Liu, Yaxun Yang, Youzhuo Wang, Xiaofei Wu, Jiamin Wang, Yichen Yao, Sören Schwertfeger, Sibei Yang, Wenping Wang, Jingyi Yu, Xuming He, and Yuexin Ma. Realdex: Towards human-like grasping for robotic dexterous hand. In *IJCAI*, 2024.

[14] Jiaxin Lu, Hao Kang, Haoxiang Li, Bo Liu, Yiding Yang, Qixing Huang, and Gang Hua. Ugg: Unified generative grasping. In *ECCV*, 2024.

[15] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, 2017.

[16] Yuzhe Qin, Yueh-Hua Wu, Shaowei Liu, Hanwen Jiang, Ruihan Yang, Yang Fu, and Xiaolong Wang. Dexmv: Imitation learning for dexterous manipulation from human videos. In *ECCV*, 2022.

[17] Stone Tao, Fanbo Xiang, Arth Shukla, Yuzhe Qin, Xander Hinrichsen, Xiaodi Yuan, Chen Bao, Xinsong Lin, Yulin Liu, Tse kai Chan, Yuan Gao, Xuanlin Li, Tongzhou Mu, Nan Xiao, Arnav Gurha, Zhiao Huang, Roberto Calandra, Rui Chen, Shan Luo, and Hao Su. Maniskill3: Gpu parallelized robotics simulation and rendering for generalizable embodied ai. *arXiv*, 2024.

[18] Dylan Turpin, Tao Zhong, Shutong Zhang, Guanglei Zhu, Eric Heiden, Miles Macklin, Stavros Tsogkas, Sven Dickinson, and Animesh Garg. Fast-grasp'd: Dexterous multi-finger grasp generation through differentiable simulation. In *ICRA*, 2023.

[19] Ruicheng Wang, Jialiang Zhang, Jiayi Chen, Yinzhen Xu, Puhao Li, Tengyu Liu, and He Wang. Dexgraspnet: A large-scale robotic dexterous grasp dataset for general objects based on simulation. In *ICRA*, 2023.

[20] Zehang Weng, Haofei Lu, Danica Kragic, and Jens Lundell. Dexdiffuser: Generating dexterous grasps with diffusion models. *RA-L*, 2024.

[21] Fanbo Xiang, Yuzhe Qin, Kaichun Mo, Yikuan Xia, Hao Zhu, Fangchen Liu, Minghua Liu, Hanxiao Jiang, Yifu Yuan, He Wang, et al. Sapien: A simulated part-based interactive environment. In *CVPR*, 2020.

[22] Yinzhen Xu, Weikang Wan, Jialiang Zhang, Haoran Liu, Zikang Shan, Hao Shen, Ruicheng Wang, Haoran Geng, Yijia Weng, Jiayi Chen, et al. Unidexgrasp: Universal robotic dexterous grasping via learning diverse proposal generation and goal-conditioned policy. In *CVPR*, 2023.

[23] Jianglong Ye, Jiashun Wang, Binghao Huang, Yuzhe Qin, and Xiaolong Wang. Learning continuous grasping function with a dexterous hand from human demonstrations. *RA-L*, 2023.

[24] Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. In Kostas E. Bekris, Kris Hauser, Sylvia L. Herbert, and Jingjin Yu, editors, *RSS*, 2023.

[25] Tony Z Zhao, Jonathan Tompson, Danny Driess, Pete Florence, Kamyar Ghasemipour, Chelsea Finn, and Ayzaan Wahid. Aloha unleashed: A simple recipe for robot dexterity. In *CoRL*, 2024.

[26] Yi Zhao, Le Chen, Jan Schneider, Quankai Gao, Juho Kannala, Bernhard Schölkopf, Joni Pajarinen, and Dieter Büchler. Rp1m: A large-scale motion dataset for piano playing with bi-manual dexterous robot hands. *arXiv preprint arXiv:2408.11048*, 2024.